

# GraphDIVE: Graph Classification by Mixture of Diverse Experts

Fenyu Hu<sup>1,2\*</sup>, Liping Wang<sup>1,2\*</sup>, Qiang Liu<sup>1,2</sup>, Shu Wu<sup>1,2†</sup>, Liang Wang<sup>1,2</sup> and Tieniu Tan<sup>1,2</sup>

<sup>1</sup> Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences

<sup>2</sup> University of Chinese Academy of Sciences

{fenyu.hu}@cripac.ia.ac.cn, {wangliping2019}@ia.ac.cn,

{qiang.liu,shu.wu,wangliang,tnt}@nlpr.ia.ac.cn

## Abstract

Graph classification is a challenging research task in many applications across a broad range of domains. Recently, Graph Neural Network (GNN) models have achieved superior performance on various real-world graph datasets. Despite their successes, most of current GNN models largely suffer from the ubiquitous class imbalance problem, which typically results in prediction bias towards majority classes. Although many imbalanced learning methods have been proposed, they mainly focus on regular Euclidean data and cannot well utilize topological structure of graph (non-Euclidean) data. To boost the performance of GNNs and investigate the relationship between topological structure and class imbalance, we propose GraphDIVE, which learns multi-view graph representations and combine multi-view experts (i.e., classifiers). Specifically, multi-view graph representations correspond to the intrinsic diverse graph topological structure characteristics. Extensive experiments on molecular benchmark datasets demonstrate the effectiveness of the proposed approach.

## 1 Introduction

Graph classification aims to identify the class label of each graph in a dataset, which is a critical and challenging problem for a broad range of real-world applications, such as drug discovery, text classification, and disease diagnosis. For instance, in chemistry, a molecule can be represented as a graph, where nodes denote atoms, and edges represent chemical bonds. Correspondingly, the classification of molecular graphs can help predict target molecular properties [Hu *et al.*, 2020].

As a powerful approach to graph representation learning, Graph Neural Network (GNN) models have been widely applied in the graph classification task [Ying *et al.*, 2018; Xu *et al.*, 2019]. Despite the huge success of GNNs, we find that the performances of current GNN models are largely

\*The first two authors contributed equally to this work.

†To whom correspondence should be addressed.

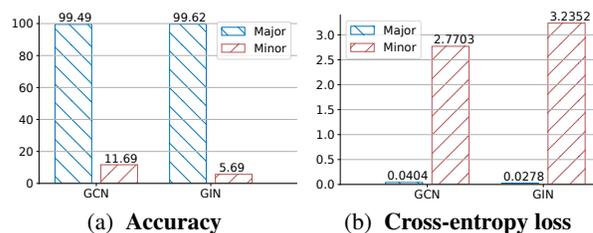


Figure 1: Test accuracy and cross-entropy loss on OGBG-HIV dataset. The low performance of minority class hinders the overall performance of GNNs.

hindered by the imbalanced class distribution, which is ubiquitous in practical applications. For example, in OGBG-MOLHIV dataset [Hu *et al.*, 2020], only about 3.5% of molecules can inhibit HIV virus replication. Figure 1 presents graph classification results of GCN [Kipf and Welling, 2017] and GIN [Xu *et al.*, 2019] on this dataset. Considering either the test accuracy or cross-entropy loss, we observe the notorious *prediction bias* phenomenon [Zhou *et al.*, 2020]: the classification performance of the minority class falls far behind that of the majority class. Apparently, the low performance of minority class hinders the overall performance of GNNs.

A straightforward solution to boost the performance of GNNs is to apply existing imbalanced learning methods, which can be divided into three categories [Liu *et al.*, 2020]: *re-sampling*, *re-weighting*, and *mixture of experts*. However, these imbalanced learning methods are initially designed for Euclidean data (such as images and texts) and they cannot model the influence of topological structure. In graph classification scenario, it is widely acknowledged that the topological structure of graphs has a significant impact on the classification performance [Xu *et al.*, 2019]. In other words, some structures might be closely related to the majority class while some other structures usually associate with the minority class. As a result, it is interesting to investigate whether and how the topological structure influences the imbalanced graph classification (IGC) problem.

To this end, we propose a novel Graph classification network with DIVerse Experts, dubbed as GraphDIVE for brevity. Among re-weighting, re-sampling, and mixture of

experts (MoE) methods, we mainly investigate MoE-based methods in this paper considering the compatibility of MoE with neural networks [Jacobs *et al.*, 1991]. To be more specific, both MoE and topological structure are related to the graph representation learning process. Therefore, we can mine the relationship between topological structure and class imbalance. In contrast, re-weighting and re-sampling is only related with the loss function and the selection of input sample, which makes it difficult to investigate the effect of topological structure. MoE is known to effectively improve typical imbalanced learning by combining the outputs of multiple experts [Dong *et al.*, 2020]. Here, each expert refers to a classifier. The success of MoE methods rely largely on the dissimilarity of experts [Kuncheva and Whitaker, 2003; Cunningham and Carney, 2000]. To further encourage the dissimilarity of experts and investigate the effect of topological structure, we propose to extract multi-view graph representations and combine multi-view experts. Specifically, when learning multi-view graph representations, the intrinsic diverse topological characteristics across the graph are also captured by GraphDIVE.

The architecture of GraphDIVE is depicted in Figure 2. At first, GraphDIVE learns diverse representations from node and graph levels. At each level, the learning process is controlled by two hyper-parameters:  $\alpha$  and  $p$ , where  $\alpha$  controls the effect of topological structure and  $p$  controls the distribution of the graph representation. By setting different values to these two hyper-parameters, we can obtain diverse graph representations, which are fed into multi-view experts to make a more accurate prediction.

To sum up, the main contributions of this work are:

- To the best of our knowledge, we are probably the first to highlight the critical importance of considering class imbalance when designing GNNs for graph classification task.
- We investigate the effect of topological structure to IGC. We propose a novel graph neural network which extracts multi-view graph representations and combines multi-view experts.
- Extensive experiments on four class imbalance molecular datasets demonstrate that GraphDIVE outperforms other state-of-the-art methods.

To foster reproducible research, our code is made publicly available at <https://github.com/CRIPAC-DIG/DIVE>.

## 2 Related Work

### 2.1 Imbalance Learning

Existing imbalance learning methods can be roughly divided into three types [Liu *et al.*, 2020]: re-sampling, re-weighting, and mixture of experts.

*Re-sampling methods* try to alleviate the imbalanced class distribution issue by controlling each class’s sample frequencies. It can be achieved by over-sampling or under-sampling [Chawla *et al.*, 2002]. Nevertheless, traditional random sampling methods usually cause over-fitting in minority classes or under-fitting in majority classes. *Re-weighting methods*

generally assign different weights to different samples. However, these methods re-weight classes proportionally to the inverse of the class frequency, which tends to make optimization difficult under extremely imbalanced settings [Huang *et al.*, 2016]. Meanwhile, we notice that Pan *et al.* [Pan and Zhu, 2013] assign different weights to graphs. But they do not consider GNN for imbalanced classification, therefore not in scope of our study. *Mixture of Experts* (MoE) is a well-studied research topic, which is also usually under a different name: classifier ensembles [Dong *et al.*, 2020]. MoE is mainly based on divide-and-conquer principle, in which the problem space is first divided and then is addressed by specialized experts [Jacobs *et al.*, 1991]. Different from above existing MoE methods that rely on a single view of representation, GraphDIVE is specially designed for graph data and explicitly explores diverse multi-view graph representations.

## 3 Preliminary

### 3.1 Problem Description

Let  $D = \{(G_1, \mathbf{Y}_1), \dots, (G_n, \mathbf{Y}_n)\}$  denote training data, where  $G_i = (\mathbf{A}_i, \mathbf{X}_i)$  denotes a graph containing the adjacency matrix and the node attribute matrix.  $\mathbf{Y}_i$  represents the labels of  $G_i$ . The task of graph classification is to learn a mapping  $f : G_i \rightarrow \mathbf{Y}_i$ . Under imbalanced classification setting, the number of instances of majority classes is far more than that of minority classes. The imbalanced graph classification (IGC) problem exists widely in practical applications, such as drug discovery, text classification and disease diagnosis.

### 3.2 MoE-based Imbalanced Learning

MoE is established based on Divide-and-Conquer (D&C) principle. Specifically, the problem space is first partitioned stochastically into a number of subspaces, then several experts are leveraged and become specialized on each subspace [Jacobs *et al.*, 1991]. The partition process is controlled by a gating network, which is trained together with the experts. Such a D&C mechanism is formulated as:

$$p(y|\mathbf{x}; \Theta) = \sum_{z=1}^M p(y, z|\mathbf{x}; \Theta) = \sum_{z=1}^M p(z|\mathbf{x}; \Theta)p(y|z, \mathbf{x}; \Theta), \quad (1)$$

where  $\mathbf{x}$  is the learned representation of a sample and  $y$  denotes the label.  $\Theta$  denotes learnable parameters of gating network and expert networks.  $z \in \{1, 2, \dots, M\}$  is a latent variable indicating expert index, and  $M$  is the number of experts. Besides,  $\sum_{z=1}^M p(z|\mathbf{x}; \Theta) = 1$  and  $p(z|\mathbf{x}; \Theta)$  is the output of the gating network, indicating the probability of assigning  $\mathbf{x}$  to the  $z$ -th expert.  $p(y|z, \mathbf{x}; \Theta)$  represents output distribution of the  $z$ -th expert.

For imbalanced learning, since using one shared classifier will inevitably lead to prediction bias towards majority classes [Zhou *et al.*, 2020], MoE methods assign different experts for different clusters instead. Here, each cluster contains semantically-close instances, and the distinction of different clusters is achieved by the gating function. Specifically, for instances of minority classes, the gating function assigns larger weights to some certain experts. Therefore,

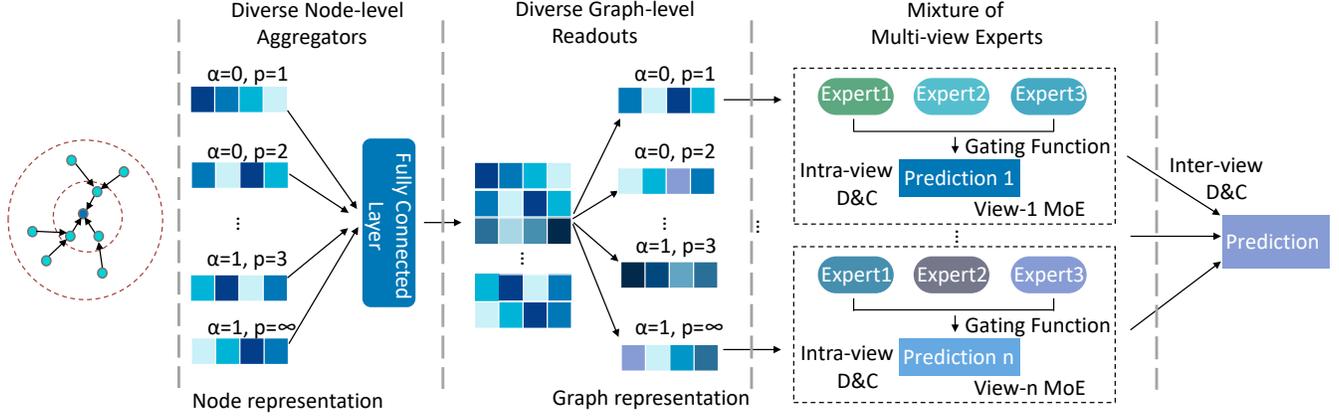


Figure 2: Model overview. GraphDIVE encourages the diversity of experts to boost the performance of GNNs. Specifically, there are mainly three components of GraphDIVE, namely, node-level aggregators, graph-level readouts, and mixture of multi-view experts. For pursuing the diversity of experts, each view of MoE is fed with a specific view of graph representation. To obtain different views of graph representations, we set different values of two hyper-parameters:  $\alpha$  and  $p$  in the first two components, respectively. In particular,  $\alpha$  controls the effect of topological structure, and  $p$  controls the distributions of representations. Finally, predictions from multi-view experts are combined, yielding the final prediction.

these experts become specialized on minority classes. Similarly, there are some other experts specialized on majority classes. As a result, the diversity of experts is a key factor for the success of MoE methods [Kuncheva and Whitaker, 2003].

## 4 Proposed Method

In this section, we present the details of the proposed Graph classification network with **DIVERSE** Experts (GraphDIVE). Existing work [Kuncheva and Whitaker, 2003; Cunningham and Carney, 2000] have found that the diversity of experts is a key factor for the success of MoE methods. Nevertheless, as formulated in Eq. (1), the experts of canonical MoE methods are based on only one shard representation, which might hinder the diversity of experts. Inspired by the development of multi-view learning [Xu *et al.*, 2013], we propose to learn multi-view graph representations to boost the diversity of experts, with the overview presented in Figure 2. GraphDIVE learns diverse graph representations from both node-level and graph-level. This corresponds to intrinsic diverse graph topological structure characteristics. Specifically, from node-level perspective, different nodes have different numbers of adjacent neighbors. From global level, different sub-graphs (motifs) recur in a graph with different frequencies. These sub-graphs are also combined in a rather complex way. As a result, multi-view graph representations can not only encourage the diversity of experts, but also capture diverse topological structure characteristics. We will introduce the details of GraphDIVE as follows.

### 4.1 Diverse Node-level Aggregators

To obtain diverse graph representations, we first design diverse node-level aggregators. This is reasonable, since only after diverse node-level characteristics are obtained, can distinct graph-level representations be captured. The diverse node-level aggregators are defined as follows:

$$\tilde{\mathbf{x}}_i^k = \left[ \sum_{j \in \mathcal{N}_i} |w_j^{\alpha_l} (\mathbf{x}_j^{k-1} - c_i)|^{p_l} \right]^{\frac{1}{p_l}}, \quad (2)$$

where  $\mathbf{x}_i^k$  indicates the representation of the  $i$ -th node at the  $k$ -th iteration, and  $\mathcal{N}_i$  is the set of nodes adjacent to node  $i$  as well as itself.  $w_j$  denotes the importance weight of neighbor node  $j$ , and  $\alpha_l$  is a hyper-parameter controlling the effect of  $w_j$ .  $c_i$  denotes bias and  $p_l$  controls the distributions of the output embeddings. Notably, the subscript  $l$  is used to distinguish  $\alpha$  and  $p$  from those in Eq. (4). Next, we present the approach of computing  $w_j$  and  $p_l$  in detail.

### Structure-aware Node Weighting

GraphDIVE assigns different weights to different nodes according to the topological structure. It is well known that neighbor nodes contribute differently in neighborhood aggregation process [Veličković *et al.*, 2018]. Since topological structure plays a crucial role in GNNs, we propose to weight different nodes considering topological structure. In graph theory and network science, node degree is a popular indicator that judges the importance of nodes in the graph. It is defined as the number of adjacent neighbors upon a node. Considering node centrality values may vary across orders of magnitude [Newman, 2018], we set  $w_j = \log(1 + d_j)/\delta$  to alleviate the impact of nodes with extremely dense connections. Specifically,  $\delta = \frac{1}{|\text{train}|} \sum_{i \in \text{train}} \log(d_i + 1)$  denotes the average degree of the training data.

Besides,  $\alpha_l \in \{0, 1\}$  controls the effect of importance weighting. Specifically, when  $\alpha_l$  equals to 0, GraphDIVE aggregates the information from neighborhoods without considering node importance. When  $\alpha_l$  equals to 1, the node weighting scheme is activated.

### Exploring Diverse Information from Neighborhoods

To capture diverse node-level characteristics of graphs, GraphDIVE also facilitates the exploring of diverse information from the neighborhoods. Each kind of information corresponds to each specific distribution in the embedding space.

As formulated in Eq. (2), the hyper-parameter  $p_l > 0$  controls the distributions of the output embeddings. When  $p_l = 1$ , this aggregator behaves as average pooling and  $p_l = \infty$  leads to max-pooling results. When  $p_l = 2$  and  $c_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{x}_j$ , it generates standard deviation of the node embeddings. Besides,  $p_l \geq 3$  enables GraphDIVE to explore other distributions from the neighborhoods.

By setting  $T$  different values to  $\alpha_l$  and  $p_l$ , we can get multiple aggregated node representations:  $\widetilde{\mathbf{x}}_i^{(1)}, \dots, \widetilde{\mathbf{x}}_i^{(T)}$ . Then these diverse representations are concatenated and fed into a fully connected (FC) layer that yields the updated node representation as:

$$\mathbf{x}_i = \text{FC}(\widetilde{\mathbf{x}}_i^{(1)} \parallel \dots \parallel \widetilde{\mathbf{x}}_i^{(T)}), \quad (3)$$

where  $\parallel$  is the concatenation operation. By doing so, we not only enrich node representation by explicitly exploring diverse node-level topological structure information, but also model the complex relations among these information.

In fact, the above diverse node-level aggregators resemble those in PNA [Corso *et al.*, 2020]. However, GraphDIVE is distinct from PNA in both motivation and technique. In motivation, GraphDIVE targets at alleviating prediction bias in the context of imbalance graph classification. It learns diverse graph representations to boost the diversity of MoE experts. In contrast, PNA focuses on distinguishing two graphs while ignoring the class imbalance problem. Technically, GraphDIVE uses multiple readouts and diverse experts, while PNA generates only one representation for each graph and uses one classifier.

### 4.2 Diverse Graph-level Readouts

In addition to promoting the diversity of node-level representations, we further encourage the diversity of graph-level representations that are generated from multiple readout functions. This is crucial, as only one graph representation is not able to fully capture the intrinsic diverse characteristics across the graph. Besides, only one graph representation may limit the diversity of experts' prediction results, hindering the performance of MoE methods. For such cases, we propose diverse graph-level readout functions as:

$$\mathbf{x} = \left[ \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \left[ w_i^{\alpha_g} (\mathbf{x}_i - c) \right]^{p_g} \right]^{\frac{1}{p_g}}, \quad (4)$$

where  $\mathbf{x} \in \mathbb{R}^{1 \times d}$  is the global graph representation, and  $d$  denotes the hidden dimension of graph embedding.  $\mathcal{N}$  denotes the set of nodes in the graph and  $w_i$  is the importance weight of node  $i$ .  $\alpha_g$  and  $p_g$  are hyper-parameters which are similar to those defined in Eq. (2). Apparently, this readout function is similar to the node-level aggregator in form. That is, each node  $i$  in the graph is weighted by  $w_i$  according to topology structure, and the weighting effect is controlled by

$\alpha_g$ . Besides,  $p_g$  controls the distributions of the output graph embeddings. By setting different values to  $\alpha_g$  and  $p_g$ , diverse graph representations can be obtained. The difference is that the node-level aggregator gathers information from local neighborhoods, while the graph-level readout function generates representation from all nodes of the graph.

From the perspective of multi-view learning [Xu *et al.*, 2013], diverse graph-level readouts generate multi-view representations for each graph. Each pair of  $\alpha_g$  and  $p_g$  corresponds to a particular view, and each view of the representation may contain some knowledge that other views do not have. Therefore, multiple views of representations can describe the graphs comprehensively and complementarily.

### 4.3 Mixture of Multi-view Experts

Based on multi-view graph representations, GraphDIVE applies multi-view experts and sends each view of graph representation to a specific view of experts. Each view of experts refers to a specific group of classifiers. In this manner, the output of different views of experts tend to be distinct, which will boost the performance of MoE. Formally, GraphDIVE generates predictions by combining the results from multi-view experts:

$$p(y|\mathbf{x}; \Theta) = \underbrace{\sum_{i=1}^K p(i|\mathbf{x}; \Theta)}_{\text{inter-view divide and conquer}} \underbrace{\sum_{z=1}^M p(z|\mathbf{x}^{(i)}; \Theta) p(y|z, \mathbf{x}^{(i)}; \Theta)}_{\text{intra-view divide and conquer}}, \quad (5)$$

where  $\mathbf{x}^{(i)}$  is the extracted graph representation from the  $i$ -th view.  $\Theta = \{\mathbf{W}_{inter} \in \mathbb{R}^{d \times K}, \mathbf{W}_{intra} \in \mathbb{R}^{d \times M}, \mathbf{W}_e \in \mathbb{R}^{d \times C}\}$  denotes learnable parameters of inter-view gating network, intra-view gating network, and expert networks respectively.  $K$ ,  $M$ , and  $C$  denote the number of views, experts, and label categories. Compared to conventional MoE methods in Eq. (1), the key distinction for GraphDIVE is that the outer summation formula combines the predictions of multi-view experts. Specifically, conventional MoE methods make predictions based on one shared (single view) representation while GraphDIVE applies different groups (multi-view) of experts on different views of representations. As shown in [Blum and Mitchell, 1998], the independence of different views can serve as a helpful complement to the multi-view learning. Applying specific experts for each individual representation promotes the independence of different views and encourages the diversity of different experts.

Since we implement each view of experts with the same network design, we drop the superscript of view index for notation simplicity and bear in mind that  $\mathbf{x}$ ,  $\mathbf{W}_{inter}$ ,  $\mathbf{W}_{intra}$  and  $\mathbf{W}_e$  are different in different view indices. First, we implement each expert with one separate fully connected layer followed by a sigmoid function:

$$p(y|z, \mathbf{x}; \Theta) = \sigma(\mathbf{x} \mathbf{W}_e^{(z)}), \quad (6)$$

where  $\mathbf{W}_e^{(z)} \in \mathbb{R}^{d \times C}$  denotes the parameters for the  $z$ -th expert.

Then, in each view of experts, the gating network generates an input-dependent soft partition of the dataset based on cosine similarity between graph representations and gating parameters:

$$p(z|\mathbf{x}; \Theta) = \frac{e^{\mathbf{x}\mathbf{W}_{intra}[z]/\tau}}{\sum_{j=1}^M e^{\mathbf{x}\mathbf{W}_{intra}[j]/\tau}}, \quad (7)$$

where  $\tau$  is the temperature hyper-parameter tuning the distribution of the gating function, and  $\mathbf{W}_{intra}[j] \in \mathbb{R}^{d \times 1}$  is the  $j$ -th column of  $\mathbf{W}_{intra}$ , which represents the gating parameter for the  $j$ -th expert.

Considering the predictions obtained in different views are based on different graph representations, they should have different contributions in judging the graph label. As such, we compute a weight score for each view as:

$$p(i|\mathbf{x}; \Theta) = \frac{e^{\mathbf{x}^{(i)}\mathbf{W}_{inter}[i]/\gamma}}{\sum_{j=1}^K e^{\mathbf{x}^{(j)}\mathbf{W}_{inter}[j]/\gamma}}, \quad (8)$$

where  $\gamma$  is a hyper-parameter that tunes the distribution.  $\mathbf{W}_{inter}[i] \in \mathbb{R}^{d \times 1}$  is the  $i$ -th column of  $\mathbf{W}_{inter}$ , which represents learnable gating parameters for each view.

#### 4.4 Model Optimization

To encourage the difference of experts in the  $i$ -th view, we introduce a Kullback–Leibler (KL) divergence regularization term as:

$$\mathcal{L}_i = -\frac{1}{M-1} \sum_{j \neq z} D_{\text{KL}}(p(y|\mathbf{x}, z; \Theta) \| p(y|\mathbf{x}, j; \Theta)), \quad (9)$$

where  $D_{\text{KL}}(\cdot, \cdot)$  is the KL divergence of two distributions.

Then, the final loss function can be formulated as:

$$\mathcal{L} = -\sum_{i=1}^K [p(i|\mathbf{x}; \Theta) \sum_{z=1}^M p(z|\mathbf{x}; \Theta) \log p(y|\mathbf{x}, z; \Theta) + \lambda \mathcal{L}_i], \quad (10)$$

where  $\lambda$  is a hyper-parameter that controls the extent of regularization.

## 5 Experiments

### 5.1 Datasets and Implementation Details

**Datasets.** We conduct experiments on four benchmark molecular property prediction datasets [Hu *et al.*, 2020], including HIV, PCBA, BACE, and BBBP. Each graph in molecular graph datasets represents a molecule, where nodes are atoms, and edges are chemical bonds. Each node contains a 9-dimensional attribute vector, including atomic number and chirality, as well as other additional atom features such as formal charge and whether the atom is in the ring.

**Implementation Details.** For a fair comparison, we implement our method and all baselines in the same experimental settings as [Hu *et al.*, 2020]. Specifically, we follow the original scaffold train-validation-test split with the ratio of 80/10/10. We run ten times for each experiment with random seed ranging from 0 to 9, and report the mean and standard deviation of test ROC-AUC for all datasets except PCBA.

Dataset	# Graphs	Avg. Size	Metric	Positive Ratio (%)
HIV	41127	25.5	ROC-AUC	3.5
PCBA	437929	26.0	AP	1.4
BACE	1513	34.1	ROC-AUC	45.6
BBBP	2039	24.1	ROC-AUC	23.5

Table 1: Statistics of molecular datasets.

Following the practice in [Hu *et al.*, 2020], we report average precision for PCBA dataset.

We evaluate the performance of the proposed GraphDIVE method on the molecular property prediction task that is a typical graph classification application. We compare with the following strong and representative GNN methods: GCN [Kipf and Welling, 2017], GIN [Xu *et al.*, 2019], FLAG [Kong *et al.*, 2020], GSN [Bouritsas *et al.*, 2020] WEGL [Kolouri *et al.*, 2021], and PNA [Corso *et al.*, 2020]. For all these methods, we use official implementation and follow the original setting.

In addition, we compare GraphDIVE with state-of-the-art imbalanced learning methods that are initially designed for Euclidean data, including FocalLoss [Lin *et al.*, 2017], LDAM [Wallach *et al.*, 2020], GHM [Li *et al.*, 2019], and Decoupling [Kang *et al.*, 2019]. The first three methods belong to re-weighting strategy and the last one belongs to re-sampling strategy. Since there is no GNN based method which considers the IGC problem, we combine these imbalanced learning methods with the representative GCN.

For hyper-parameter setting, we train the model using Adam optimizer [Kingma and Ba, 2015] with initial learning rate of 0.001. For HIV and PCBA datasets, we train the network for 200 epochs in light of the scale of the dataset. Moreover, for all the other datasets, we train the model for 100 epochs. According to the average performance on the validation dataset, we use grid-search to find the optimal value for  $K$  (i.e., the number of views),  $M$  (i.e., the number of experts), and  $\lambda$ . We set the hyper-parameter space of  $K$  and  $M$  as  $\{2, 3, 4, 5, 6, 7, 8\}$  and the hyper-parameter space of  $\lambda$  as  $\{0.001, 0.01, 0.1, 1, 10\}$ , respectively. Besides, the hyper-parameter space of  $\alpha$  is  $\{0, 1\}$  and the hyper-parameter space of  $p$  is  $\{1, 2, 3, +\infty\}$ . The hyper-parameter space of  $\tau$  and  $\gamma$  is  $\{0.001, 0.01, 0.1, 1, 10, 100\}$ .

### 5.2 Molecular Property Prediction

**Comparison with other GNNs.** For molecular property prediction task, we report classification results of state-of-the-art GNN models in Table 2. Overall, GraphDIVE consistently outperforms other GNN models across all four datasets. For example, PNA [Corso *et al.*, 2020] is one of the latest state-of-the-art GNN methods for graph classification, and we can observe that GraphDIVE achieves 0.68%, 0.44%, 3.19%, and 1.23% absolute improvement on HIV, PCBA, BACE, and BBBP dataset, respectively.

**Comparison with other imbalanced learning methods.** We also compare GraphDIVE with other state-of-the-art imbalanced learning methods in Table 2. Firstly, it can be observed that the state-of-the-art imbalanced learning methods, such as LDAM and Decoupling, do not seem to offer significant or stable improvements over GNN models. For example,

	HIV	PCBA	BACE	BBBP
GCN	76.06±0.97	20.20±0.24	79.15±1.44	68.87±1.51
GIN	75.58±1.4	22.66±0.28	72.97±4.00	68.17±1.48
GCN+FLAG	76.83±1.02	21.16±0.17	80.53±1.43	70.04±0.82
GIN+FLAG	76.54±1.14	23.95±0.40	80.02±1.68	68.60±1.27
WEGL	77.57±1.11	20.52±0.35	78.06 ± 0.91	68.27 ± 0.99
GSN	77.99±1.00	19.78±0.28	76.53±4.54	67.90±1.86
PNA	79.05 ± 1.32	28.38 ± 0.35	81.85±1.68	69.13±1.72
Focal Loss (RW)	76.56±1.15	22.84±0.32	81.08±2.02	67.90±1.16
GHM (RW)	75.33±1.44	19.96±0.35	80.51±1.54	67.04±1.26
LDAM (RW)	76.58±1.69	20.48±0.27	78.91±2.10	67.08±0.94
Decoupling (RS)	78.15±1.28	24.32±0.24	80.01±1.01	68.42±1.46
GraphDIVE	<b>79.73±0.63</b>	<b>28.82±0.26</b>	<b>85.04±1.13</b>	<b>70.36±1.24</b>

Table 2: Summary of classification results (%) for imbalanced molecular property prediction. RW and RS are the abbreviation of re-weighting and re-sampling, respectively.

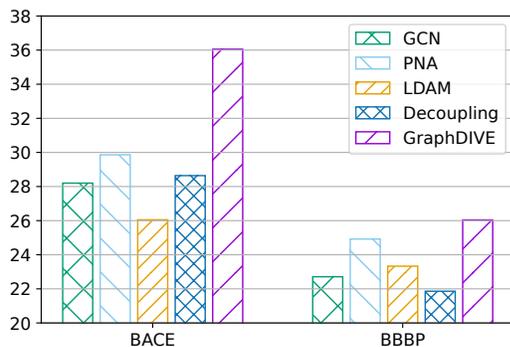


Figure 3: Test accuracy (%) of minority class on BACE and BBBP dataset.

LDAM performs better than GCN on HIV and PCBA dataset, but it is inferior to GCN on BACE and BBBP dataset. We suppose the reason is that either re-sampling or re-weighting methods make the model focus more on minority class, resulting in potential over-fitting to minority class [Zhou *et al.*, 2020]. Besides, they cannot capture the diverse topological structure either. In contrast, we observe that GraphDIVE outperforms these imbalanced learning methods by a large margin. We conduct more experiments in the following parts to investigate why GraphDIVE brings performance improvement.

### 5.3 Prediction Bias Analysis

In this subsection, we study whether the overall performance improvement of GraphDIVE is achieved by alleviating the IGC problem. Specifically, we report the classification accuracy of minority class in Figure 3.

We have the following observations. First of all, GraphDIVE outperforms GCN, PNA and existing imbalanced learning methods regarding the performance of minority class. This result demonstrates that GraphDIVE can alleviate the prediction bias. Secondly, existing state-of-the-art re-weighting and re-sampling methods, such as LDAM and Decoupling, have marginal performance improvements or even performance degradation in minority class. This may be because re-weighting and re-sampling methods cannot model the relationship between the topological structure and the class imbalance. We give a more detailed study in Sec. 5.4.

D-Node	D-Graph	MoE	ROC-AUC (%)
			79.15 ± 1.44
✓			81.85 ± 1.68
	✓		82.50 ± 0.91
		✓	82.88 ± 0.96
✓	✓		83.14 ± 0.86
✓		✓	82.76 ± 1.15
	✓	✓	83.81 ± 0.79
✓	✓	✓	<b>85.04 ± 1.13</b>

Table 3: Ablation study on the effectiveness of each component on BACE dataset. D-Node and D-Graph refer to Diverse node-level aggregators and Diverse graph-level readout functions, respectively. We apply multi-view MoE only when D-Graph is used. Otherwise, traditional single-view MoE is directly used. For any component that is not checked, we apply its corresponding component in GCN (i.e., symmetric normalized average aggregator, mean readout function, or single layer classifier) for substitute.

### 5.4 Ablation Study

To verify the effectiveness of each component, we conduct an ablation study on node-level neighborhood aggregators, graph-level readout functions and multi-view experts respectively. The results are shown in Table 3. When each component is individually applied (2nd, 3rd and 4th row), the performance is improved compared with vanilla GCN (1st row). This can be attributed to the exploring of complementary graph information and the D&C mechanism, respectively. Besides, comparing the 4th row and the last two rows, we can observe that multi-view node-level representations and graph-level representations can improve the performance of vanilla MoE methods. This is because diverse graph representations encourage the diversity of different experts.

Aside from the above results shown in Table 3, we also investigate effect of topological structure. Specifically, we discard the structure-aware node weighting scheme by setting  $\alpha_l$  and  $\alpha_g$  as zero. We observe 0.75% and 1.16% performance degradation on HIV and BACE dataset, respectively. This result demonstrates that it is necessary and beneficial to model topological structure for the IGC problem.

## 6 Conclusion

Existing GNNs largely suffer from the ubiquitous class imbalance problem. In this paper, we have proposed GraphDIVE, a graph neural network with mixture of diverse experts to alleviate the prediction bias towards majority classes. GraphDIVE learns multi-view graph representations from both node-level and graph level. These multi-view graph representations can not only encourage the diversity of experts, but also capture diverse topological structure characteristics. Experimental results on four datasets exhibit the effectiveness and generalization ability of GraphDIVE.

### Acknowledgements

This work is jointly supported by National Natural Science Foundation of China (62141608, U19B2038) and CAAI Huawei MindSpore Open Fund.

## References

- [Blum and Mitchell, 1998] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 1998.
- [Bouritsas *et al.*, 2020] Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *arXiv preprint arXiv:2006.09252*, 2020.
- [Chawla *et al.*, 2002] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 2002.
- [Corso *et al.*, 2020] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. In *Advances in Neural Information Processing Systems*, 2020.
- [Cunningham and Carney, 2000] Padraig Cunningham and John Carney. Diversity versus quality in classification ensembles based on feature selection. In *European Conference on Machine Learning*, 2000.
- [Dong *et al.*, 2020] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 2020.
- [Hu *et al.*, 2020] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, 2020.
- [Huang *et al.*, 2016] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [Jacobs *et al.*, 1991] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Comput.*, 1991.
- [Kang *et al.*, 2019] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2019.
- [Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.
- [Kolouri *et al.*, 2021] Soheil Kolouri, Navid Naderializadeh, Gustavo K. Rohde, and Heiko Hoffmann. Wasserstein embedding for graph learning. In *International Conference on Learning Representations*, 2021.
- [Kong *et al.*, 2020] Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. FLAG: adversarial data augmentation for graph neural networks. *arXiv preprint arXiv:2010.09891*, 2020.
- [Kuncheva and Whitaker, 2003] Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 2003.
- [Li *et al.*, 2019] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *AAAI Conference on Artificial Intelligence*, 2019.
- [Lin *et al.*, 2017] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [Liu *et al.*, 2020] Zhining Liu, Pengfei Wei, Jing Jiang, Wei Cao, Jiang Bian, and Yi Chang. Mesa: Boost ensemble imbalanced learning with meta-sampler. *Advances in Neural Information Processing Systems*, 2020.
- [Newman, 2018] Mark E. J. Newman. *Networks: An Introduction (Second Edition)*. Oxford University Press, 2018.
- [Pan and Zhu, 2013] Shirui Pan and Xingquan Zhu. Graph classification with imbalanced class distributions and noise. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.
- [Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *International Conference on Learning Representations*, 2018.
- [Wallach *et al.*, 2020] H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2020.
- [Xu *et al.*, 2013] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.
- [Xu *et al.*, 2019] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.
- [Ying *et al.*, 2018] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, 2018.
- [Zhou *et al.*, 2020] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.